|  |
| --- |
| **VyOS API**[](https://docs.vyos.io/en/latest/automation/vyos-api.html#vyos-api) |

|  |
| --- |
| For configuration and enabling the API see HTTP-API |

**Authentication**[](https://docs.vyos.io/en/latest/automation/vyos-api.html#authentication)

|  |  |
| --- | --- |
| All endpoints only listen on HTTP POST requests and the API KEY must set as key in the   |  | | --- | |  | |

formdata.

Below see one example for curl and one for python. The rest of the documentation is reduced to curl.

|  |
| --- |
| curl --location --request POST 'https://vyos/retrieve' \ --form data='{"op": "showConfig", "path": []}' \ --form key='MY-HTTPS-API-PLAINTEXT-KEY'  **importrequests**  url ="https://vyos/retrieve"  payload={'data': '{"op": "showConfig", "path": []}',   'key': 'MY-HTTPS-API-PLAINTEXT-KEY'  }  headers = {}  response = requests.request("POST", url, headers=headers, data=payload) print(response.text) |

**API Endpoints**[](https://docs.vyos.io/en/latest/automation/vyos-api.html#api-endpoints)

**/retrieve**[](https://docs.vyos.io/en/latest/automation/vyos-api.html#retrieve)

|  |  |  |
| --- | --- | --- |
| With the retrieve endpoint you get parts or the whole configuration.   |  | | --- | |  |   To get the whole configuration, pass an empty list to the path field   |  | | --- | |  | |

|  |
| --- |
| curl --location --request POST 'https://vyos/retrieve' \--form data='{"op": "showConfig", "path": []}' \--form key='MY-HTTPS-API-PLAINTEXT-KEY' |

|  |
| --- |
| response (shorted)  {  "success": true,  "data": {  "interfaces": {  "ethernet": {  "eth0": {  "address": "dhcp",  "duplex": "auto",  "hw-id": "50:00:00:01:00:00", |

|  |
| --- |
| "speed": "auto"  },  "eth1": {  "duplex": "auto",  "hw-id": "50:00:00:01:00:01",  "speed": "auto"  ...  },  "error": null  } |
| To only get a part of the configuration, for example system syslog .   |  | | --- | |  | |

|  |
| --- |
| curl -k --location --request POST 'https://vyos/retrieve' \ --form data='{"op": "showConfig", "path": ["system", "syslog"]}' \--form key='MY-HTTPS-API-PLAINTEXT-KEY' |

|  |  |  |
| --- | --- | --- |
| response:  {  "success": true,  "data": {  "global": {  "facility": {  "all": {  "level": "info"  },  "protocols": {  "level": "debug"  }  }  }  },  "error": null  }  if you just want the Value of a multi-valued node, use the returnValues operation.   |  | | --- | |  |   For example, get the addresses of a dum0 interface.   |  | | --- | |  | |

|  |
| --- |
| curl -k --location --request POST 'https://vyos/retrieve' \ --form data='{"op": "returnValues", "path": ["interfaces","dummy","dum0","address"]}' \--form key='MY-HTTPS-API-PLAINTEXT-KEY' |

|  |
| --- |
| respone:  {  "success": true,  "data": [  "10.10.10.10/24",  "10.10.10.11/24",  "10.10.10.12/24"  ],  "error": null  } |

**/reset**[](https://docs.vyos.io/en/latest/automation/vyos-api.html#reset)

|  |  |  |
| --- | --- | --- |
| The reset endpoint run a reset command.   |  | | --- | |  |  |  | | --- | |  | |

|  |
| --- |
| curl --location --request POST 'https://vyos/reset' \ --form data='{"op": "reset", "path": ["ip", "bgp", "192.0.2.11"]}' \--form key='MY-HTTPS-API-PLAINTEXT-KEY' |

|  |
| --- |
| respone:  {  "success": true,  "data": "",  "error": null  } |

**/image**[](https://docs.vyos.io/en/latest/automation/vyos-api.html#image)

|  |  |
| --- | --- |
| To add or delete an image, use the /image endpoint.   |  | | --- | |  | |

add an image

|  |
| --- |
| curl -k --location --request POST 'https://vyos/image' \ --form data='{"op": "add", "url": "https://downloads.vyos.io/rolling/current/amd64/vyos-rolling-latest.iso"}' \ --form key='MY-HTTPS-API-PLAINTEXT-KEY' |

|  |  |
| --- | --- |
| respone (shorted):  {  "success": true,  "data": "Trying to fetch ISO file from https://downloads.vyos.io/rolling-latest.iso\n ...  Setting up grub configuration...\nDone.\n",  "error": null  }  delete an image, for example 1.3-rolling-202006070117   |  | | --- | |  | |

|  |
| --- |
| curl -k --location --request POST 'https://vyos/image' \ --form data='{"op": "delete", "name": "1.3-rolling-202006070117"}' \--form key='MY-HTTPS-API-PLAINTEXT-KEY' |

|  |
| --- |
| response:  {  "success": true,  "data": "Deleting the \"1.3-rolling-202006070117\" image...\nDone\n", "error": null  } |

**/show**[](https://docs.vyos.io/en/latest/automation/vyos-api.html#show)

|  |  |
| --- | --- |
| The /show endpoint is to show everything in the operational mode.   |  | | --- | |  | |

For example, show which images are installed.

|  |
| --- |
| curl -k --location --request POST 'https://vyos/show' \ --form data='{"op": "show", "path": ["system", "image"]}' \--form key='MY-HTTPS-API-PLAINTEXT-KEY' |

|  |
| --- |
| response:  {  "success": true,  "data": "The system currently has the following image(s) installed:\n\n 1: 1.4-rolling-202102280559 (default boot)\n  2: 1.4-rolling-202102230218\n  3: 1.3-beta-202102210443\n\n",  "error": null  } |

**/generate**[](https://docs.vyos.io/en/latest/automation/vyos-api.html#generate)

|  |  |  |
| --- | --- | --- |
| The generate endpoint run a generate command.   |  | | --- | |  |  |  | | --- | |  | |

|  |
| --- |
| curl -k --location --request POST 'https://vyos/generate' \ --form data='{"op": "generate", "path": ["wireguard", "default-keypair"]}' \--form key='MY-HTTPS-API-PLAINTEXT-KEY' |

|  |
| --- |
| response:  {  "success": true,  "data": "",  "error": null  } |

**/configure**[](https://docs.vyos.io/en/latest/automation/vyos-api.html#configure)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| You can pass a set , delete or comment command to the /configure endpoint.   |  | | --- | |  |  |  | | --- | |  |  |  | | --- | |  |  |  | | --- | |  |  |  | | --- | |  |   set a single command |

|  |
| --- |
| curl -k --location --request POST 'https://vyos/configure' \ --form data='{"op": "set", "path": ["interfaces", "dummy", "dum1", "address", "10.11.0.1/32"]}' \--form key='MY-HTTPS-API-PLAINTEXT-KEY' |

|  |  |
| --- | --- |
| response:  {  "success": true,  "data": null,  "error": null  }   |  | | --- | |  |   delete a single command |

|  |
| --- |
| curl -k --location --request POST 'https://vyos/configure' \ --form data='{"op": "delete", "path": ["interfaces", "dummy", "dum1", "address", "10.11.0.1/32"]}' \--form key='MY-HTTPS-API-PLAINTEXT-KEY' |

|  |
| --- |
| response:  {  "success": true,  "data": null,  "error": null  } |

The API pushes every request to a session and commit it. But some of VyOS components like DHCP and PPPoE Servers, IPSec, VXLAN, and other tunnels require full configuration for commit. The endpoint will process multiple commands when you pass them as a list to

|  |  |
| --- | --- |
| the data field.   |  | | --- | |  | |

|  |
| --- |
| curl -k --location --request POST 'https://vyos/configure' \ --form data='[{"op": "set","path":["interfaces","vxlan","vxlan1","remote","203.0.113.99"]}, {"op": "set","path":["interfaces","vxlan","vxlan1","vni","1"]}]' \ --form key='MY-HTTPS-API-PLAINTEXT-KEY' |

|  |
| --- |
| response:  {  "success": true,  "data": null,  "error": null  } |

**/config-file**[](https://docs.vyos.io/en/latest/automation/vyos-api.html#config-file)

|  |  |
| --- | --- |
| The endpoint /config-file is to save or load a configuration.   |  | | --- | |  | |

Save a running configuration to the startup configuration. When you don’t specify the file when

|  |  |
| --- | --- |
| saving, it saves to /config/config.boot .   |  | | --- | |  | |

|  |
| --- |
| curl -k --location --request POST 'https://vyos/config-file' \--form data='{"op": "save"}' \ --form key='MY-HTTPS-API-PLAINTEXT-KEY' |

|  |
| --- |
| response:  {  "success": true,  "data": "Saving configuration to '/config/config.boot'...\nDone\n", "error": null  } |

Save a running configuration to a file.

|  |
| --- |
| curl -k --location --request POST 'https://vyos/config-file' \--form data='{"op": "save", "file": "/config/test.config"}' \--form key='MY-HTTPS-API-PLAINTEXT-KEY' |

|  |
| --- |
| response:  {  "success": true,  "data": "Saving configuration to '/config/test.config'...\nDone\n", "error": null  } |

To Load a configuration file.

|  |
| --- |
| curl -k --location --request POST 'https://vyos/config-file' \--form data='{"op": "load", "file": "/config/test.config"}' \--form key='MY-HTTPS-API-PLAINTEXT-KEY' |

|  |
| --- |
| response:  {  "success": true,  "data": null,  "error": null  } |